AWS REAL TIME PROJECT-2

AUTOMATED FILE SYNCHRONIZATION WITH AWS S3 USING SHELL SCRIPTING AND CORN

The overview of the project

In this Project, we explore the power of automation using shell scripting in conjunction with

the AWS command line interference (CLI). The objective is to create a robust local files with

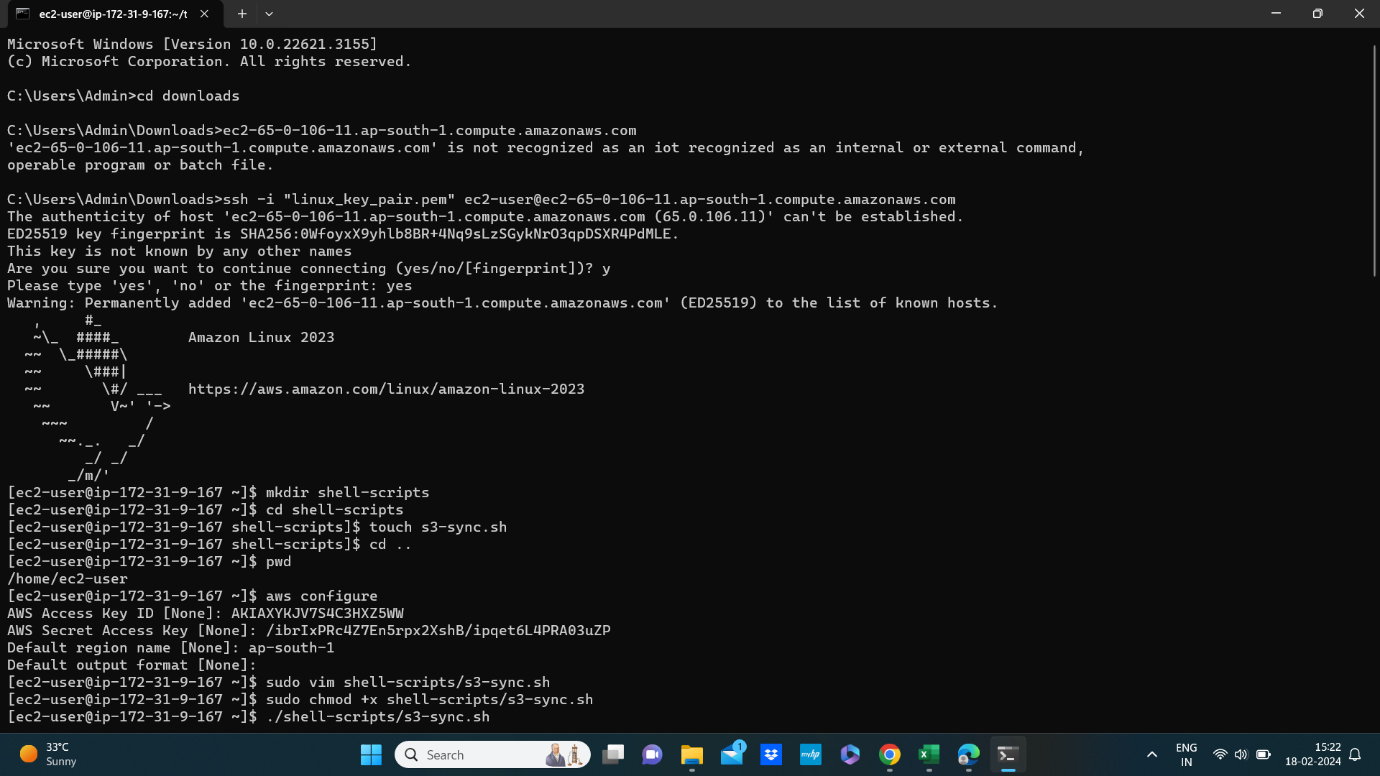
an Amazon S3 bucket. To make this process even more hands-free, we will utilize cron jobs

to schedule these sync operations at specific intervals.

And the technologies used for this project is

* AWS CLI : this allow for command line access to various amazon web services
* Shell-scripting: using bash scripting to crate a script that automates the file synchronization process
* Corn : it is time based job scheduler in unix like operating system

1. **Connect to your Amazon Linux instance**: Make sure you're connected to your Amazon Linux instance via SSH or your preferred method.



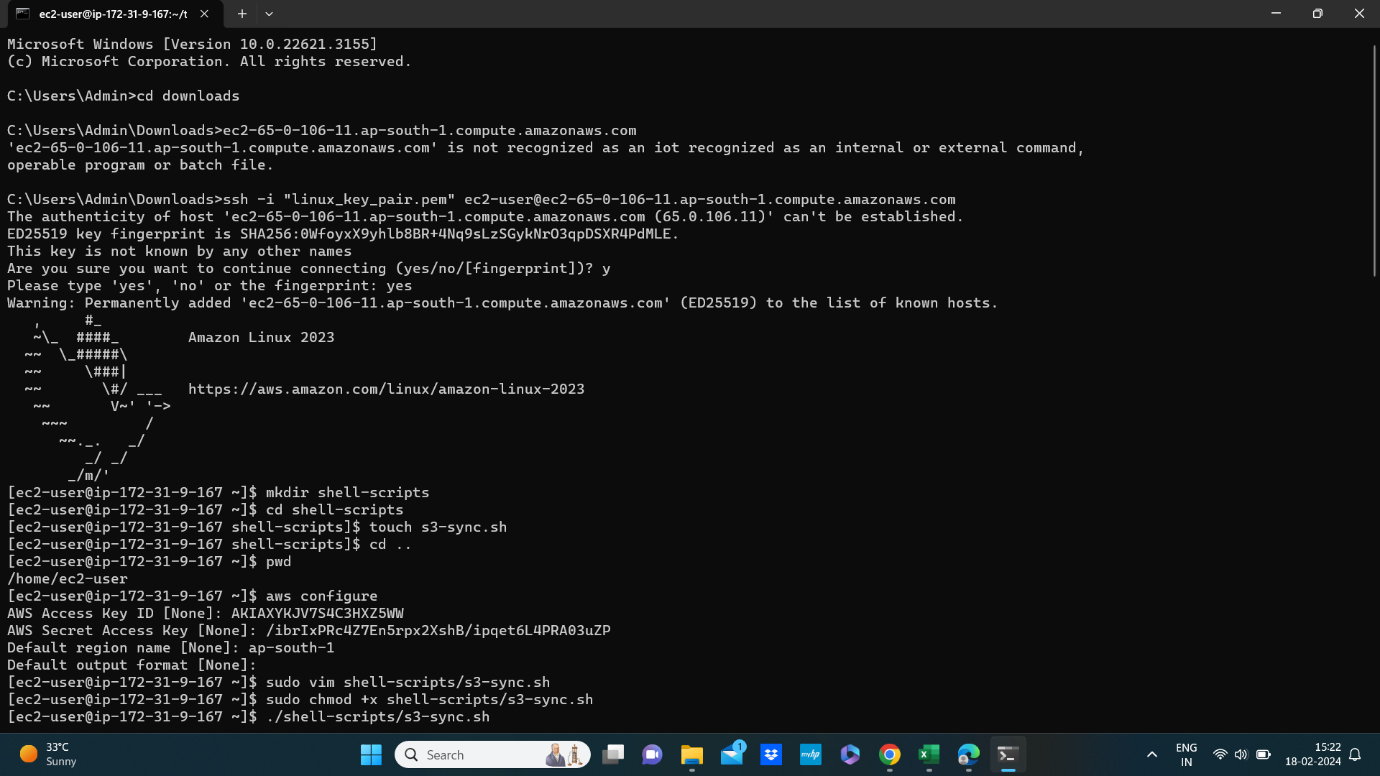
C:\users\admin>cd downloads

ssh -i “linix\_keu\_pair.perm” [ec2-user@ec2-65-0-106-11.ap-south-1.compute.amazons.com](mailto:ec2-user@ec2-65-0-106-11.ap-south-1.compute.amazons.com)

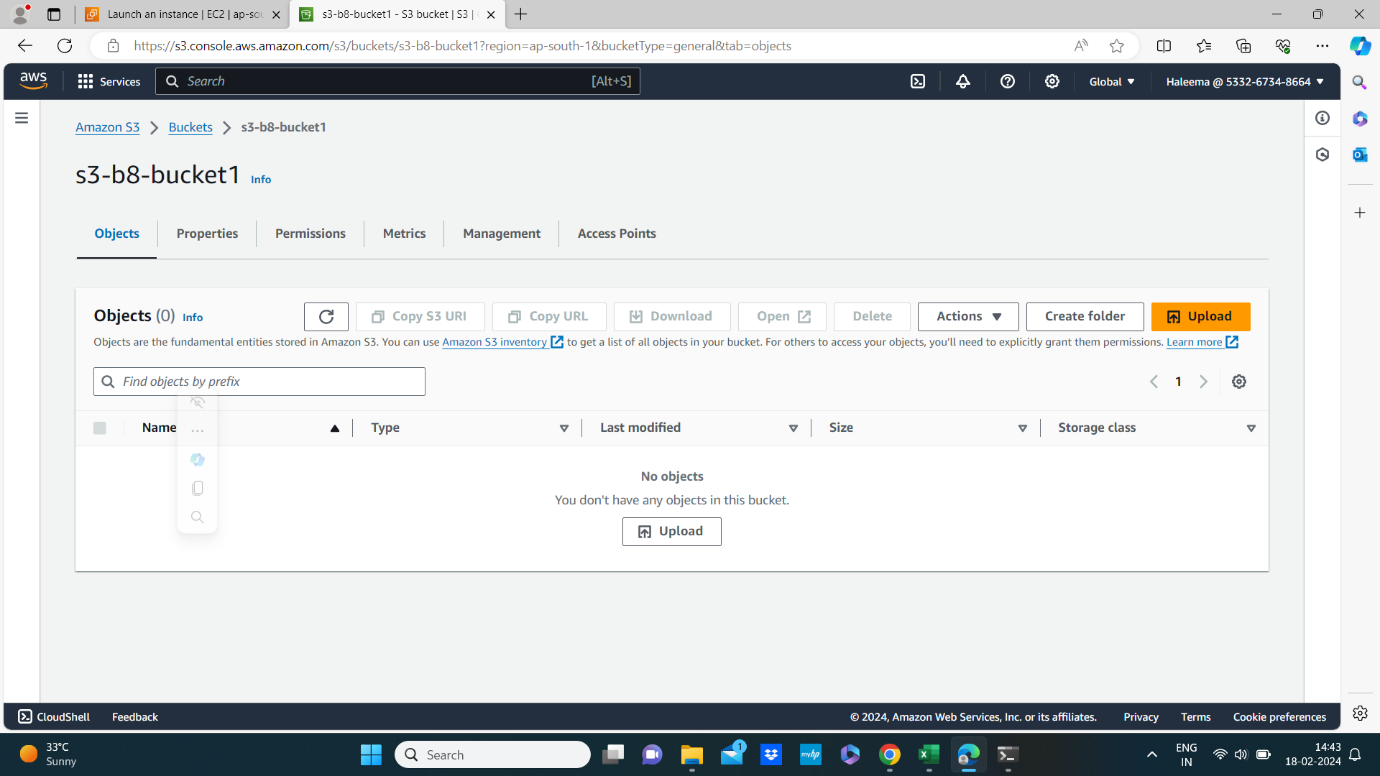
1. **Navigate to the directory containing the script**: Use the **cd** command to navigate to the directory where your shell script is located. For example:

* Make directory: Mkdir shell-scripts
* Change directory: Cd shell-scripts
* Create a folder: touch s3-sync.sh
* Pwd: /home/ec2-user

1. **Aws configure**: configure the AWS CLI with your AWS credentials using the **aws configure** command. You'll need your AWS Access Key ID, Secret Access Key, region, and preferred output format.

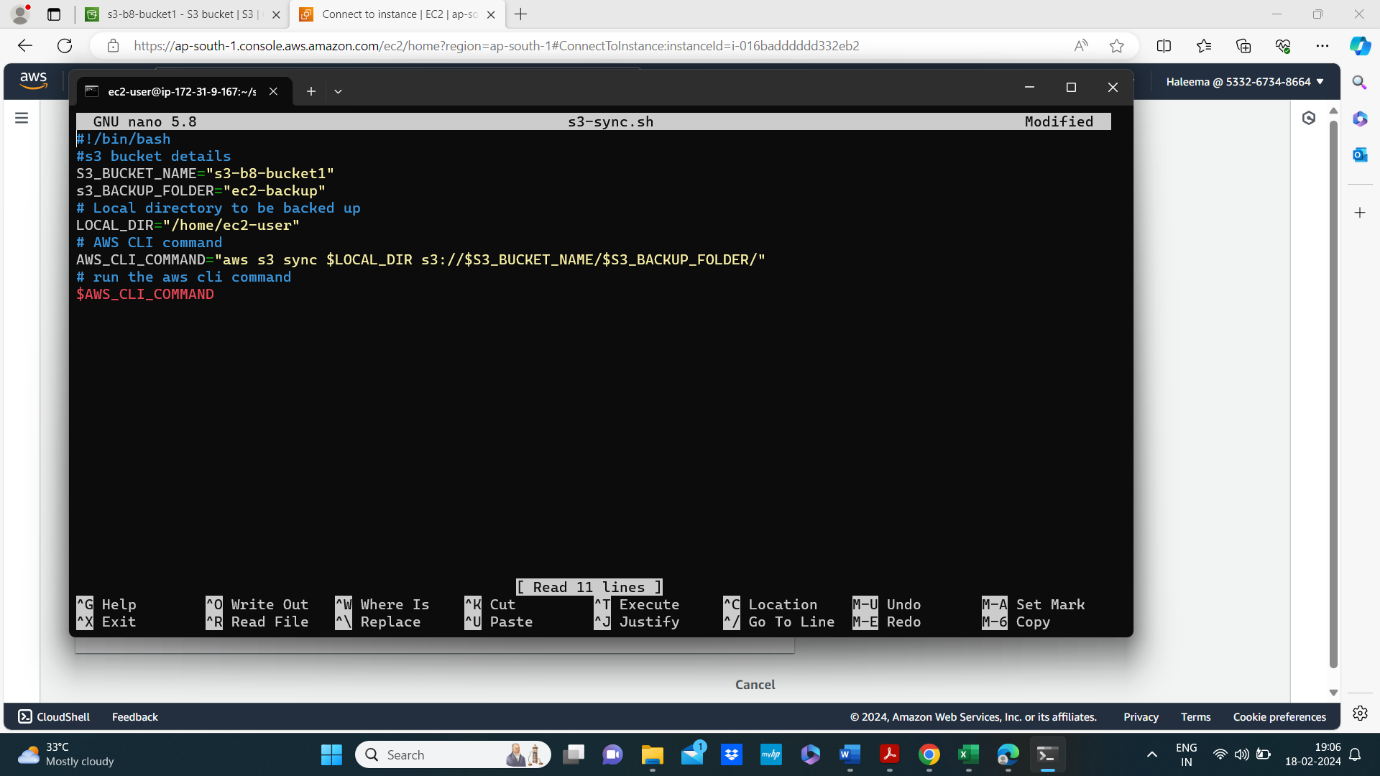


1. **Create s3 bucket:**



Since AWS CLI interacts with AWS services, you don't navigate through directories as you would on your local machine. Instead, you specify the S3 path to the directory or file you want to work with in each command.

1. **A bash script in the directory:** You can edit a bash script stored in an S3 bucket using a text editor on your local machine. You can download the script using the **aws s3 cp** command, edit it using a text editor of your choice, and then upload it back to the same location using **aws s3 cp** again.



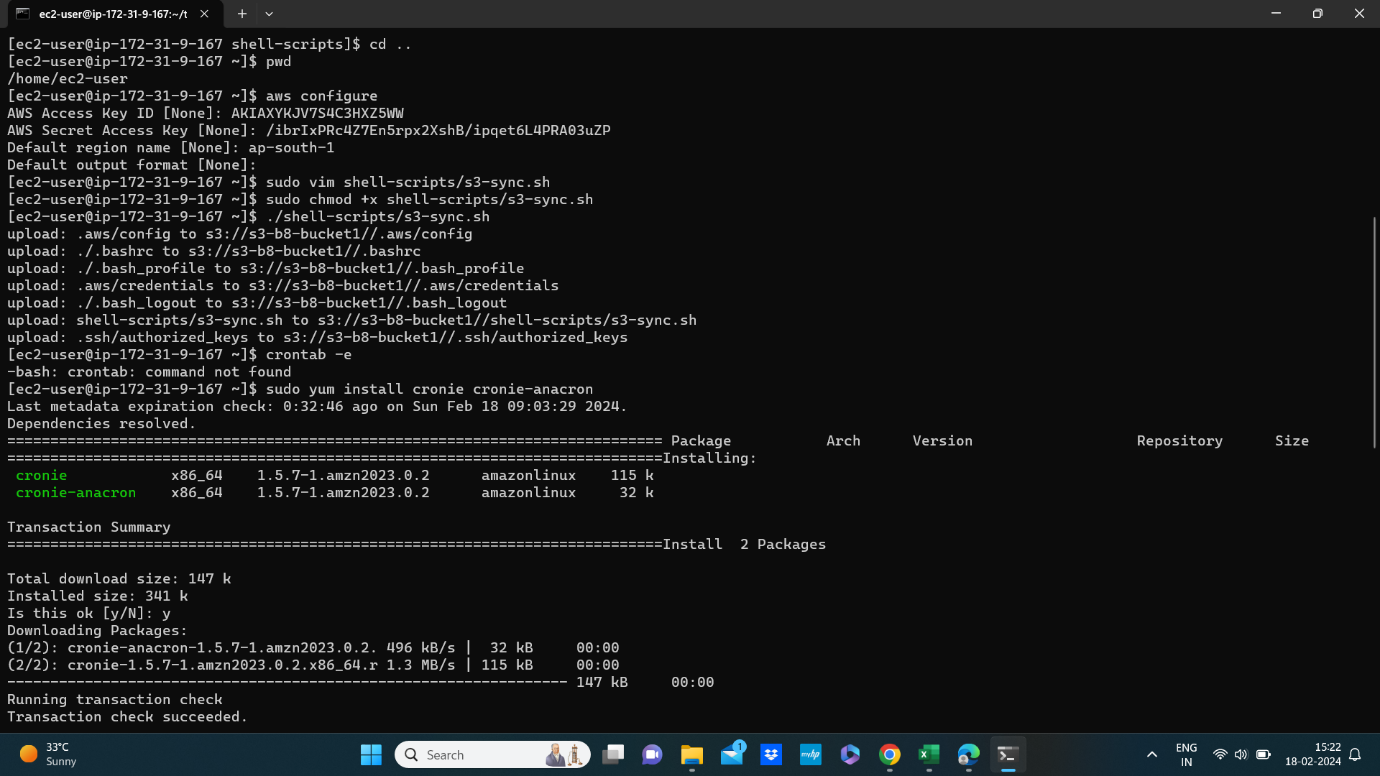
**6.Make the script executable (if necessary):** If you want to execute the script directly, you may need to make it executable using the **chmod** command:

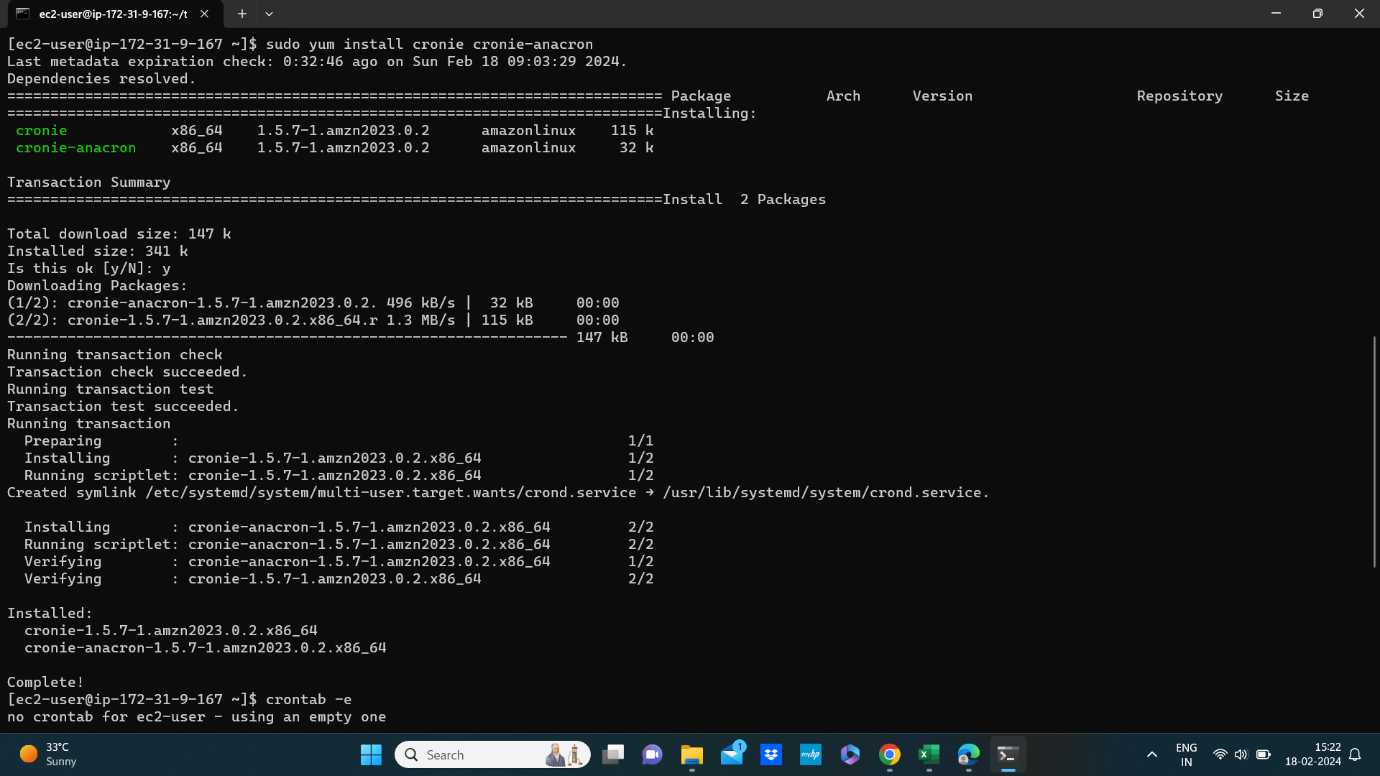
**chmod +x s3-sync.sh**

This command gives execute permissions to the owner of the file (**+x**).

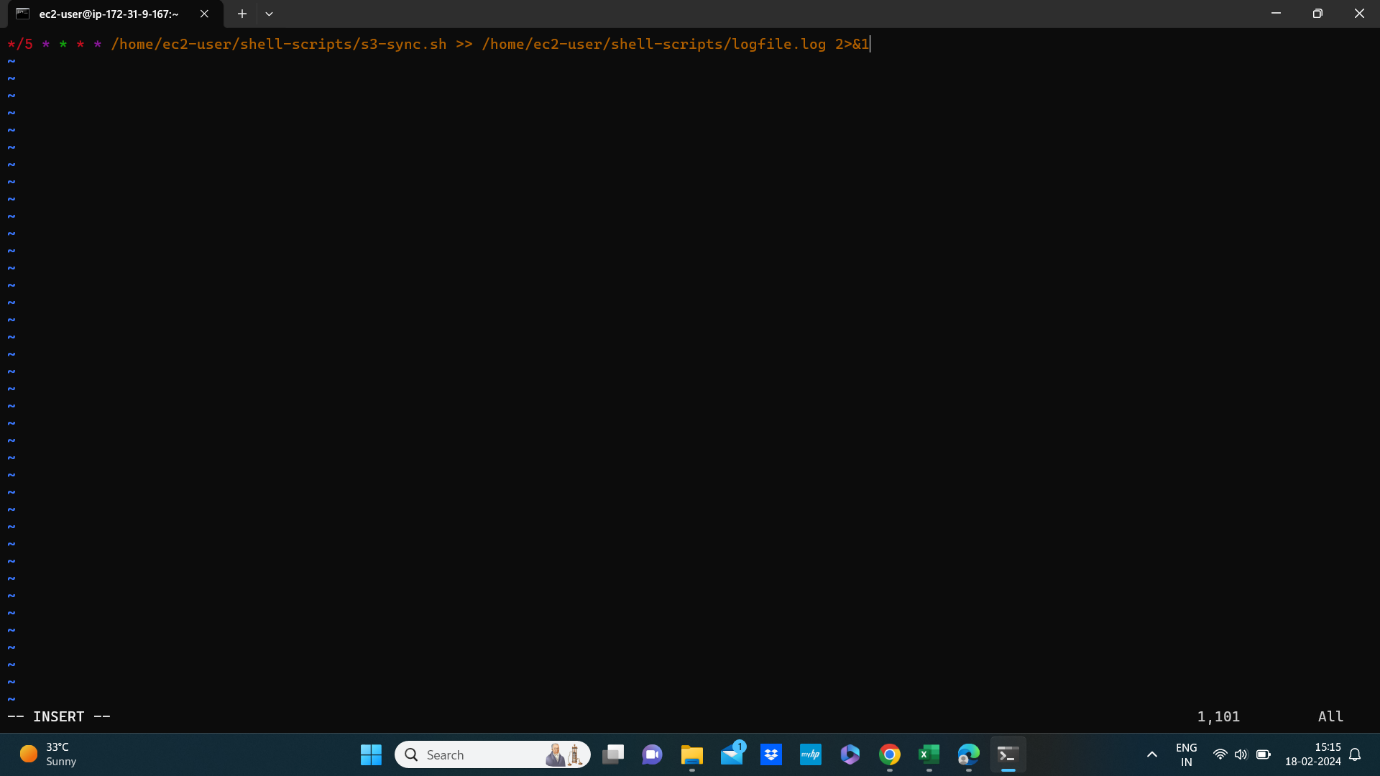
That's it! You've created a bash script in Amazon Linux using the AWS CLI (by navigating and editing with shell commands). You can now execute the script using **./s3-sync.sh** if you made it executable.

**7.Schedule with Cron**: Use cron to schedule the execution of the script at specific intervals. Open your crontab file by running **crontab -e** and add an entry like this:





**8.Monitor and Troubleshoot**: Regularly monitor the cron job execution and the synchronization process. Check the logfile for any errors or unexpected behavior.



By following these steps, you should be able to automate file synchronization between your local directory and an AWS S3 bucket using shell scripting and cron.

